

İŞLETİM SİSTEMLERİNE GİRİŞ

find komutu

- ▶ Zaman zaman adının yalnızca bir kısmını bildiğiniz ama bulunduğu dizini bir türlü hatırlayamadığınız dosyalar ve dizinler olacaktır.
- ▶ Tek tek bütün dizinlere girip ls komutuyla bu dosya ya da dosyaları aramaktansa find komutunu kullanmak çok daha kullanışlı olacaktır.
- ▶ `find /home/nur -name bilgisayar`
- ▶ `find /etc -name *.conf`
- ▶ `find /etc -name *apache*`

```
nur@pardus:~/Masaüstü$ find /home/nur/ -name sıralama
/home/nur/Masaüstü/sıralama
nur@pardus:~/Masaüstü$ find /etc -name *.conf
/etc/libreoffice/psprint.conf
/etc/cups/cupsd.conf
/etc/cups/snmp.conf
/etc/cups/cups-browsed.conf
```

ps komutu

- ▶ Sistem yöneticisinin sıklıkla kullandığı en önemli komutlardan bir tanesidir.
- ▶ Çalışmakta olan processleri (süreç) listeler.
- ▶ -aux parametresi ile çok sık kullanılır.
- ▶ `ps -aux | grep nur`
- ▶ (Üstteki komut sonrasında sadece nur kullanıcısına ait süreçler ekranda listelenir.)

clear

- Konsol ekranını temizler.

file

- file isim
- İsim adlı elemanın cinsini gösterir. Dosyaysa dosyanın türünü (txt, exe, jpeg, gif, vb.), dizinse dizin olduğunu söyler.

```
nur@pardus:~/Masaüstü$ file sıralama
sıralama: UTF-8 Unicode (with BOM) text
nur@pardus:~/Masaüstü$ file siir
siir: UTF-8 Unicode text
nur@pardus:~/Masaüstü$ file 2
2: OpenDocument Spreadsheet
nur@pardus:~/Masaüstü$
```

history

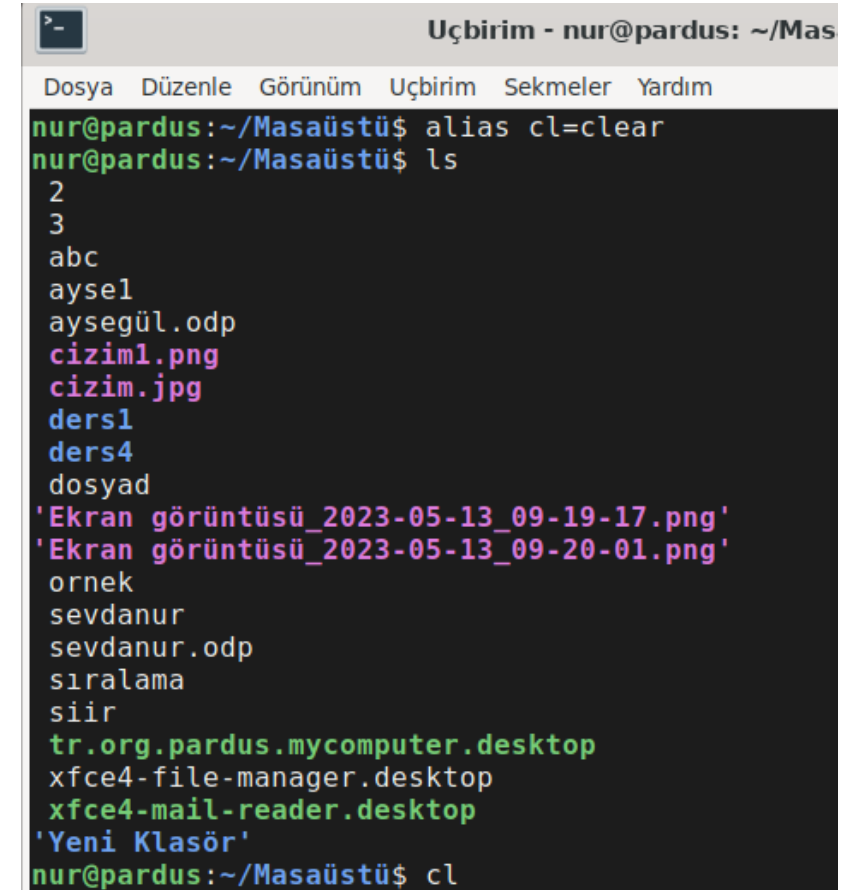
- ▶ Daha önce konsolda kullanılmış olan komutları listeler.
- ▶ history 20
- ▶ Son kullanılan 20 komutu listeler.
- ▶ Tekrar kullanmak istediğiniz komut için ! işareti ve komutun önündeki sayıyı yazarak kullanılır.
- ▶ 1443. sırada clear varsa komut satırına !1443 yazmak yeterli

```
nur@pardus:~/Masaüstü$ history
 1  xfce4-settings-manager~
 2  xfce4-settings-manager~
 3  sudo xfce4-settings-manager~
 4  mugshot
 5  blueman-adapters
 6  mugshot
 7  usermod
 8  usermode
 9  usermod sevda
10  sudo apt install filelight
11  sudo apt update && sudo apt upgrade
12  echo $SHELL
13  cp --help
14  whatis help
15  whatis cp
16  whatis man
17  whatis cd
18  whatis cd..
19  whatis pwd
20  whatis mkdir
21  ls
22  ls -a
```

```
Uçbirim - nur@pardus: ~/Masaüstü
Dosya  Düzenle  Görünüm  Uçbirim  Sekmeler  Yardım
177  grep sevda sıralama
178  grep sevda /home/nur/Belgeler
179  clear
180  grep sevda sıralama
181  grep sevda /home/nur/Masaüstü/
182  clear
183  grep sevda /home/nur/Masaüstü/
184  clear
185  grep sevda sıralama
186  grep sevda /home/nur/Masaüstü/siir
187  grep -v sevda sıralama
188  grep -n sevda /home/nur/Masaüstü/
189  clear
190  grep -v sevda sıralama
191  grep -n sevda /home/nur/Masaüstü/*
192  find /home/nur/ -name sıralama
193  find /etc -name *.conf
194  clear
195  ps -aux
196  file sıralama
197  file siir
198  file 2
199  history
nur@pardus:~/Masaüstü$ !194
```

alias

- ▶ Komuta takma ad verme
- ▶ Alias komutu ile sık kullandığımız komutlara isim verebiliriz.
- ▶ Alias liste='ls -la'
- ▶ Komutuyla ls -la komutunu liste takma adıyla tanımladık.
- ▶ unalias liste
- ▶ liste takma adını kaldırır.

A terminal window titled 'Uçbirim - nur@pardus: ~/Mas' showing a series of commands and their outputs. The user sets an alias 'cl=clear' and then lists files in the current directory. The output shows various files including images, documents, and desktop shortcuts. Finally, the user runs 'cl' which clears the terminal screen.

```
>_ Uçbirim - nur@pardus: ~/Mas
Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
nur@pardus:~/Masaüstü$ alias cl=clear
nur@pardus:~/Masaüstü$ ls
2
3
abc
ayse1
aysegül.odp
cizim1.png
cizim.jpg
ders1
ders4
dosyad
'Ekrana görüntüsü_2023-05-13_09-19-17.png'
'Ekrana görüntüsü_2023-05-13_09-20-01.png'
ornek
sevdanur
sevdanur.odp
siralama
siir
tr.org.pardus.mycomputer.desktop
xfce4-file-manager.desktop
xfce4-mail-reader.desktop
'Yeni Klasör'
nur@pardus:~/Masaüstü$ cl
```

Dosya Sıkıştırma Yöntemleri

- ▶ Zip ile sıkıştırma
- ▶ `zip -r deneme.zip dosyaAdı/dizinAdı`
- ▶ Zip dosyasını açmak
- ▶ `unzip deneme.zip`

```
nur@pardus:~/Masaüstü$ zip -r dersnotları.zip dersnotları
adding: dersnotları/ (stored 0%)
adding: dersnotları/Linux2002.pdf (deflated 15%)
nur@pardus:~/Masaüstü$
```


- ▶ **rar ile sıkıştırma (*)**
- ▶ **rar a -ap deneme.rar dosya/dizin**

- ▶ **rar dosyasını açmak (*)**
- ▶ **unrar e deneme.rar**
- ▶ **unrar x deneme.rar**

Kullanıcı oluşturma

- ▶ **sudo su**
 - ▶ Root kullanıcısına geçiş yapar.
- ▶ **Adduser kullanıcı_adi**
 - ▶ Yeni kullanıcı oluşturur. Sadece root yeni kullanıcı oluşturabilir.
- ▶ **addgroup grup_adi**
 - ▶ Yeni grup oluşturur. Sadece root yeni grup oluşturabilir.
- ▶ **passwd kullanıcı_adi**
 - ▶ Kullanıcının şifresini değiştirir.

Kullanıcı silme

- ▶ **su kullanıcı_adi**
- ▶ Kullanıcı_adi isimli kullanıcıya geçer.

- ▶ **deluser kullanıcı_adi**
- ▶ Adı verilen kullanıcıyı siler. Bu yetki root'a aittir.

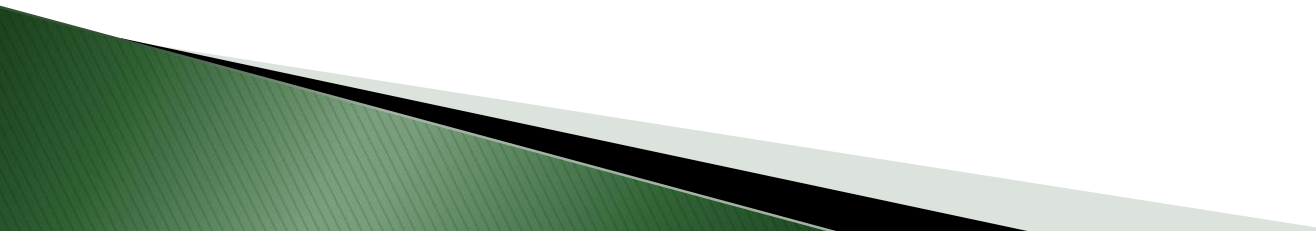
- ▶ **deluser -r kullanıcı_adi**
- ▶ Kullanıcı ile beraber kullanıcıya ait ev dizini de silinir.

- ▶ **delgroup grup_adi**
- ▶ Adı verilen grubu siler. Bu yetki sadece root'a aittir.

Sistemle ilgili

- ▶ **who**
 - ▶ Sisteme bağlı olan kullanıcıları listeler.
- ▶ **whoami**
 - ▶ Sisteme giriş ismini gösterir.
- ▶ **hostname**
 - ▶ Linux server kurulu sunucumuzun tam ismini görüntüler.
- ▶ **ifconfig**
 - ▶ ip adresi ve Ethernet kartı ile ilgili bilgi (Ethernet kartının ismi, Mac adresi, netmask adresi vb) verir. Root yetkisiyle çalışır

- ▶ **last**
 - ▶ Sisteme ne zaman giriş yapılmış ne zaman reboot edilmiş gösterir.

 - ▶ **uptime**
 - ▶ Sistemin ne kadar zamandır açık onu gösterir.
- 

- ▶ **uname**
- ▶ İşletim sistemi hakkında bilgi verir.
- ▶ **-m**
- ▶ Makinanın donanımının adını verir.
- ▶ **-r**
- ▶ Kernelinizin (çekirdek) sürümünü görmek için
- ▶ **-s**
- ▶ Sistem ismini yazar
- ▶ **-v**
- ▶ İşletim sisteminin versiyonunu yazar
- ▶ **-a**
- ▶ yukarıdaki bilgilerin tamamını yazar

- ▶ **echo Komutu**
- ▶ Karakterleri ya da değişkenlerin değerlerini görüntüler. Kullanıcının kabuk programı, sistemde **\$SHELL** adlı değişkende saklanır. Bu değişkenin değerini görüntülemek için, aşağıda gösterildiği gibi **echo** komutu kullanılabilir.

```
nur@pardus:~$ echo $SHELL
/bin/bash
nur@pardus:~$ kim="sevda duran"
nur@pardus:~$ echo kim
kim
nur@pardus:~$ echo $kim
sevda duran
nur@pardus:~$ echo "hosgeldin kim"
hosgeldin kim
nur@pardus:~$ echo "hosgeldin $kim"
hosgeldin sevda duran
nur@pardus:~$
```

- **cmp Komutu (Compare)**
- İki dosyayı karşılaştırarak farklılıklarını belirtir.

```
nur@pardus:~/Masaüstü$ cd /home/nur/Masaüstü
nur@pardus:~/Masaüstü$ cat siir
Düş bir yaş dalından düşerse
Nereye düşer hiç düşündünüz mü?
Yerde bir iz kalmayacak mı izdüşüm?
Düşen yaş dalından düşünce
Gözlerinizdedir pınarı
Bir yaş bir daldan düşünce
Kökündedir yaşı
Bir yaş düşer bir daldan
Hepimizin ölen arkadaşı
Ve çok eskilere dair bir düşünce
Can Yücel
nur@pardus:~/Masaüstü$ cat sıralama
beyza
sevda
nur
ali
alya
ayse
kadir
umut
ali
sevda nur duran
nur@pardus:~/Masaüstü$ cmp siir sıralama
siir sıralama farklı: bayt 1, satır 1
nur@pardus:~/Masaüstü$
```


- ▶ **diff Komutu (Different)**
- ▶ İki metin dosyasını karşılaştırır. Farklılıkları ayrıntılı olarak gösterir.

```
nur@pardus:~/Masaüstü$ diff siir siralama
1,11c1,10
< Düş bir yaş dalından düşerse
< Nereye düşer hiç düşündünüz mü?
< Yerde bir iz kalmayacak mı izdüşüm?
< Düşen yaş dalından düşünce
< Gözlerinizdedir pınarı
< Bir yaş bir daldan düşünce
< Kökündedir yaşı
< Bir yaş düşer bir daldan
< Hepimizin ölen arkadaşı
< Ve çok eskilere dair bir düşünce
< Can Yücel
\ Dosya sonunda yenisatır yok.
---
> beyza
> sevda
> nur
> ali
> alya
> ayse
> kadir
> umut
> ali
> sevda nur duran
nur@pardus:~/Masaüstü$
```

- ▶ **expr Komutu (Evaluate Expressions)**
- ▶ Aritmetiksel ve mantıksal işlemleri gerçekleştirir.

```
nur@pardus:~$ expr 3 + 5
8
nur@pardus:~$ expr 15 / 4
3
nur@pardus:~$ expr 16 / 4
4
nur@pardus:~$ expr 1 == 4
0
nur@pardus:~$ expr 1 == 1
1
```

► bc Komutu

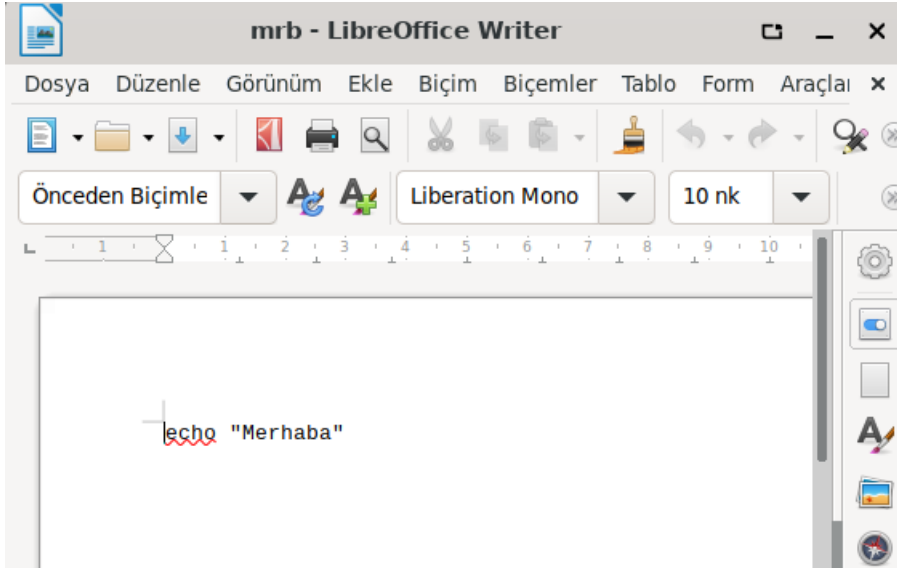
- Aritmetik işlemlerin gerçekleştirildiği bir programdır. Ondalıkli sayılarla çalışmaya olanak sağlar.

```
nur@pardus:~/Masaüstü$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006,
n, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
5.2 + 2.3
7.5
```

Temel Kabuk Programlama

- ▶ **Kabuk Programlamaya Giriş**
- ▶ Kabuk komutları, bir dosya içerisine yazılarak, bu komutların ardarda çalışması sağlanabilir. Bu komutların girdi ve çıktıları birbirleriyle ilişkilendirilerek belirli işlevleri gerçekleştirirler. Bu yapı ile oluşturulan ve hazırlanan komut dizisi, kabuk programı (shell script) olarak adlandırılır.

- Ev dizininde “mrb” adlı bir dosya oluşturarak içerisine aşağıdaki komutu yazalım.



Daha sonra Konsol uygulamasını açalım ve dosyaya **chmod u+x** komutuyla çalıştırma hakkı verelim.

```
nur@pardus:~$ chmod u+x mrb
nur@pardus:~$
```

chmod u+x
filename
sadece dosyanın
sahibine
çalıştırma izni
verilir.

Bu adımlardan sonra dosyayı aşağıdaki şekilde çalıştırabilirsiniz.

```
nur@pardus:~$ chmod u+x mrb
nur@pardus:~$ ./mrb
Merhaba
nur@pardus:~$
```

Yazılan programa çalıştırma hakkı vermeden de **sh** komutuyla çalıştırabilirsiniz.

```
nur@pardus:~$ sh mrb
Merhaba
nur@pardus:~$
```

- ▶ **Değişkenler**
- ▶ Değişkenler (variables), bir programın çalışması boyunca, gerekli olan verileri saklayan ve gerektiğinde bunları değiştirmeye olanak sağlayan yapılardır.
- ▶ Değişkenler, Linux kabuğunda “=” ile tanımlanabilir.

```
nur@pardus:~$ kullanıcı_adi="sevda"  
nur@pardus:~$ kullanıcı_mail="sevda@gmail.com"  
nur@pardus:~$ echo $kullanıcı_adi  
sevda  
nur@pardus:~$ echo $kullanıcı_mail  
sevda@gmail.com  
nur@pardus:~$
```

“kullanıcı_adi” ve “kullanıcı_mail” isminde iki adet değişken tanımlanmıştır. Daha sonra echo komutuyla bu değişkenlerin değerleri yazdırılmıştır.

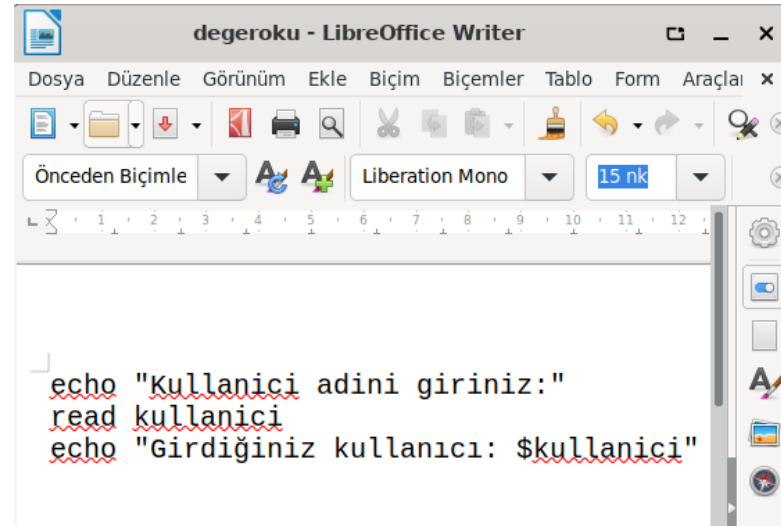
Değişkenler tanımlanırken “=” ile ifadeler arasında boşluk bulunmamasına dikkat ediniz. Değişken tanımlanırken değişken adının önünde herhangi bir sembol bulunmaz; fakat program içerisinde kullanırken değişken isimlerinin önüne \$ sembolünün getirilmesi gerekir.

► Değer Okuma

- Değişkenlere programın çalışması sırasında dışarıdan değer atanabilir. Bu işlem, **read** komutu ile gerçekleştirilir.

Dosya adı “**degeroku**” olacak bir metin editörü oluşturalım.

Daha sonra içerisine çalıştırmak istediğimiz program kodlarını yazalım.



```
nur@pardus:~$ chmod u+x degeroku  
nur@pardus:~$ ./degeroku  
Kullanıcı adını giriniz:  
sevdanur  
Girdiğiniz kullanıcı: sevdanur  
nur@pardus:~$
```


► Kontrol Yapıları

- Kontrol işlemlerinde **test** komutu, **if-else** ve **case** deyimleri kullanılır.

► test Komutu

- test komutu, mantıksal bir ifadenin sonucunu bulur. Bash kabuğunda en son çalışan komutun sonucu \$? ile öğrenilebilir. Eğer komut başarılı bir şekilde çalışmışsa 0 (sıfır) sonucunu, diğer durumlarda sıfırdan farklı bir değeri geri döndürür. Bu özellikten yararlanarak test komutunun sonucu da ekranda görüntülenebilir.

```
nur@pardus:~$ test 5 -gt 3
nur@pardus:~$ echo $?
0
nur@pardus:~$ test 5 -gt 6
nur@pardus:~$ echo $?
1
```

Aritmetik	
-eq	Eşit
-gt	Büyük
-lt	Küçük
-ge	Büyük eşit
-le	Küçük eşit

Yukarıdaki örnekte test komutu, **-gt (greater than)** seçeneğiyle kullanılmıştır. Bu seçenek ile önce 5 sayısının 3'ten büyük olup olmadığı test edilmiş ve komut hatasız çalışarak sonuçta 0 (sıfır) bulunmuştur. Diğer test işleminde ise 5 sayısının 6'dan büyük olup olmadığı test edilmiş, 5 sayısı 6 dan büyük olmadığından sonuç hatalı bir şekilde 1 (bir) olarak geri döndürülmüştür.

► If-Else Deyimi

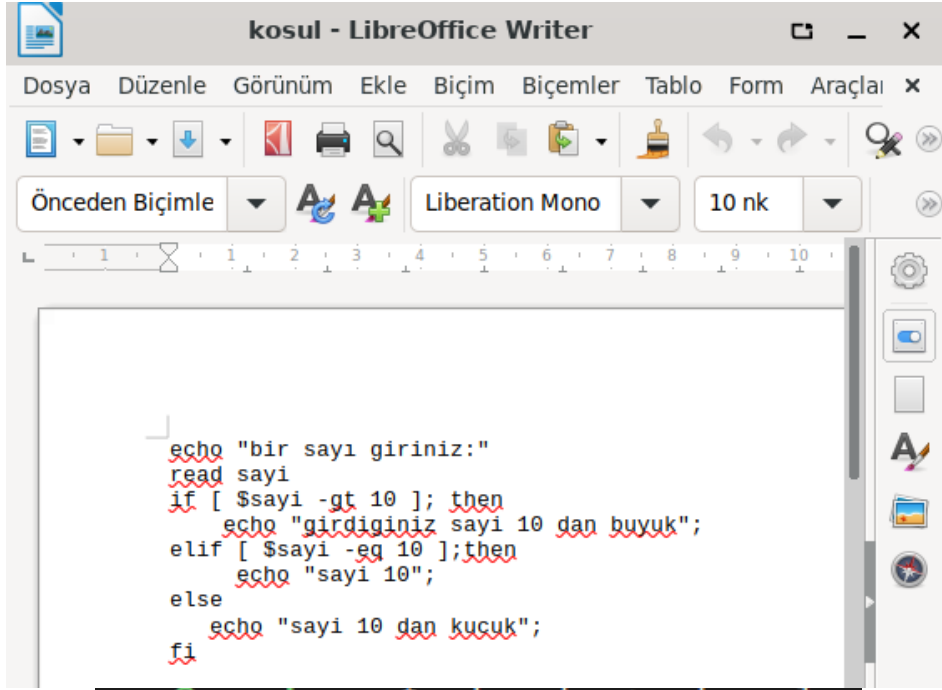
- If deyimi, mantıksal bir ilişkiyi sınayarak bir işlemin yapılmasına, yapılmamasına ya da başka bir işlemin gerçekleştirilmesine karar vermek için kullanılan deyimdir. If, “eğer” anlamındadır.

```
if [ koşul-1 ]; then
    koşul-1 doğru ise yürütülecek komutlar
elif [ koşul-2 ]; then
    koşul-2 doğru ise yürütülecek komutlar
.....
elif [ koşul-n ]; then
    koşul-n doğru ise yürütülecek komutlar
else
    hiçbir koşul doğru değilse yürütülecek komutlar
fi
```

if deyimi, deyimi oluşturan harflerin tersten yazılmasıyla oluşan **fi komutu** ile sonlandırılır.

Bir if deyiminde, **if komutu**, ilk **koşul**, **then ifadesi**, **koşul doğruysa çalışacak komut** ve **fi komutu** zorunludur.

Diğer kısımlar, gereksinimlere bağlı olarak kullanılabilir.



```
kosul - LibreOffice Writer
Dosya  Düzenle  Görünüm  Ekle  Biçim  Biçemler  Tablo  Form  Araçlar x
Önceden Biçimle  Liberation Mono  10 nk
1 2 3 4 5 6 7 8 9 10
echo "bir sayı giriniz:"
read sayi
if [ $sayi -gt 10 ]; then
    echo "girdiginiz sayi 10 dan büyük";
elif [ $sayi -eq 10 ];then
    echo "sayi 10";
else
    echo "sayi 10 dan küçük";
fi
```

```
nur@pardus:~$ chmod u+x kosul
nur@pardus:~$ ./kosul
bir sayı giriniz:
12
girdiginiz sayi 10 dan büyük
nur@pardus:~$ ./kosul
bir sayı giriniz:
5
sayi 10 dan küçük
nur@pardus:~$ ./kosul
bir sayı giriniz:
10
sayi 10
```

if koşulunda \$sayi değişkeninin değerinin 10'dan büyük olması durumu sınanmaktadır.

Koşul doğru olduğunda 5. satırdaki komut çalışır. elif kısmında 10'a eşit olma durumuna bakılmaktadır. Bu koşul doğru olursa, 7. satırdaki komut çalışır. else kısmında değişken değeri yukarıdaki koşullara uymadığında (değişken değerinin 10'dan küçük olması durumunda) 9. satırda çalışacak komut belirtilmiştir.

Aritmetik	
-eq	Eşit
-gt	Büyük
-lt	Küçük
-ge	Büyük eşit
-le	Küçük eşit

► Case Deyimi

- Bir veya birden fazla koşula bağlı olarak birçok seçenek arasında seçim yapılması gerektiğinde case deyimi kullanılır.

```
case değişken in  
    seçenek-1)  
        komutlar  
    ;;  
    seçenek-2)  
        komutlar  
    ;;  
    ...  
    *)  
        komutlar  
    ;;  
esac
```

* seçeneği, üstteki koşullardan herhangi biri gerçekleşmediğinde çalışır.

```
~/secim - Mousepad
Dosya Düzenle Arama Görünüm Belge Yardım

echo "İslemler Menusu"
echo "1-Ekranı temizle"
echo "2-Ev dizinini listele"
echo "3-Yönetici girisi yap"
echo "Secenek seciniz (1-3):"
read secenek

case $secenek in
1)
    clear
;;
2)
    ls -l ~
;;
3)
    su nur ~
;;
*)
    echo "Gecersiz secim"
esac
```

```
nur@pardus:~$ chmod u+x secim
nur@pardus:~$ ./secim
İslemler Menusu
1-Ekranı temizle
2-Ev dizinini listele
3-Yönetici girisi yap
Secenek seciniz (1-3):
2
toplam 56
drwxr-xr-x 5 nur nur 4096 May 20 09:54 Belgeler
-rwxr--r-- 1 nur nur 89 Haz 2 09:53 degeroku
drwxr-xr-x 2 nur nur 4096 Mar 25 00:42 Genel
drwxr-xr-x 2 nur nur 4096 May 5 17:23 İndirilenler
-rwxr--r-- 1 nur nur 187 Haz 2 18:02 kosul
drwxr-xr-x 6 nur nur 4096 May 26 14:06 Masaüstü
-rwxr--r-- 1 nur nur 24 May 31 23:54 merhaba
-rwxr--r-- 1 nur nur 14 Haz 2 09:55 mrb
drwxr-xr-x 2 nur nur 4096 Mar 25 00:42 Müzik
drwxr-xr-x 2 nur nur 4096 Mar 25 00:42 Resimler
drwxr-xr-x 2 nur nur 4096 May 5 16:59 Scratch
-rwxr--r-- 1 nur nur 246 Haz 2 18:17 secim
drwxr-xr-x 2 nur nur 4096 Mar 25 00:42 Şablonlar
drwxr-xr-x 2 nur nur 4096 Mar 25 00:42 Videolar
```

```
nur@pardus:~$ ./secim
İslemler Menusu
1-Ekranı temizle
2-Ev dizinini listele
3-Yönetici girisi yap
Secenek seciniz (1-3):
3
Parola:
/home/nur: /home/nur: Bir dizin
```

► Döngüler

► while Döngüsü

- Belirli işlemlerin, bir koşulun varlığında tekrarlanması istendiğinde kullanılan yapıdır. while, iken anlamındadır. while'in yanında belirtilen koşul doğru iken yapı içerisindeki komutların tekrarlamasını sağlar. Koşul yanlış olduğunda döngü sonlanır, **done** ifadesinden sonraki komutlara geçilir.

```
while koşul  
do  
    komutlar  
done
```

```
~/dongu - Mousepad
Dosya Düzenle Arama Görünüm Belge Yardım
sayac=1
while [ $sayac -le 10 ]
do
    echo $sayac
    (( sayac = $sayac + 1 ))
done
```

Aritmetik	
-eq	Eşit
-gt	Büyük
-lt	Küçük
-ge	Büyük eşit
-le	Küçük eşit

```
nur@pardus:~$ chmod u+x dongu
nur@pardus:~$ ./dongu
```

```
1
2
3
4
5
6
7
8
9
10
```

Döngü Sayısı	sayac Önceki Değer	sayac Yeni Değer
İlk durum	-	1
1. Döngü	1	2
2. Döngü	2	3
3. Döngü	3	4
4. Döngü	4	5
5. Döngü	5	6
6. Döngü	6	7
7. Döngü	7	8
8. Döngü	8	9
9. Döngü	9	10
10. Döngü	10	11
Döngüye girilmez	11	-

► for Döngüsü

1. Kullanım:

```
for (( ifade1 ; ifade2 ; ifade3 ))  
do  
    ifade2 yanlış olana kadar  
    do ve done arasındaki tüm  
    komutları tekrarlar.  
Done
```

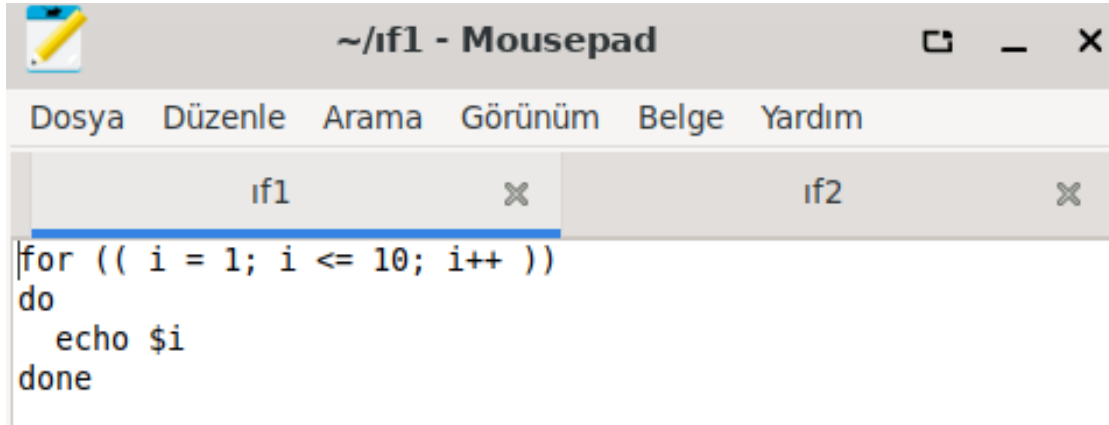
ifade1: Döngü sayaç değişkeninin ilk değeri

ifade2: Döngü koşulu

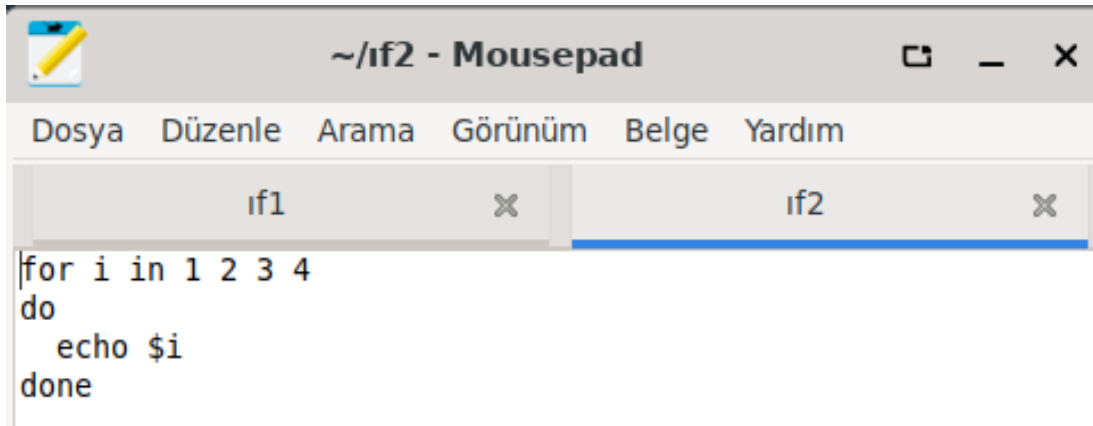
ifade3: Sayaç değişkeninin değiştirildiği aritmetik ifade

2. Kullanım:

```
for değişken in değerler  
do  
    komutlar  
done
```



```
~/if1 - Mousepad
Dosya  Düzenle  Arama  Görünüm  Belge  Yardım
if1    x        if2    x
for (( i = 1; i <= 10; i++ ))
do
  echo $i
done
```



```
~/if2 - Mousepad
Dosya  Düzenle  Arama  Görünüm  Belge  Yardım
if1    x        if2    x
for i in 1 2 3 4
do
  echo $i
done
```

```
nur@pardus:~$ chmod u+x if1
nur@pardus:~$ ./if1
1
2
3
4
5
6
7
8
9
10
nur@pardus:~$ chmod u+x if2
nur@pardus:~$ ./if2
1
2
3
4
```

► Kabuk Fonksiyonları

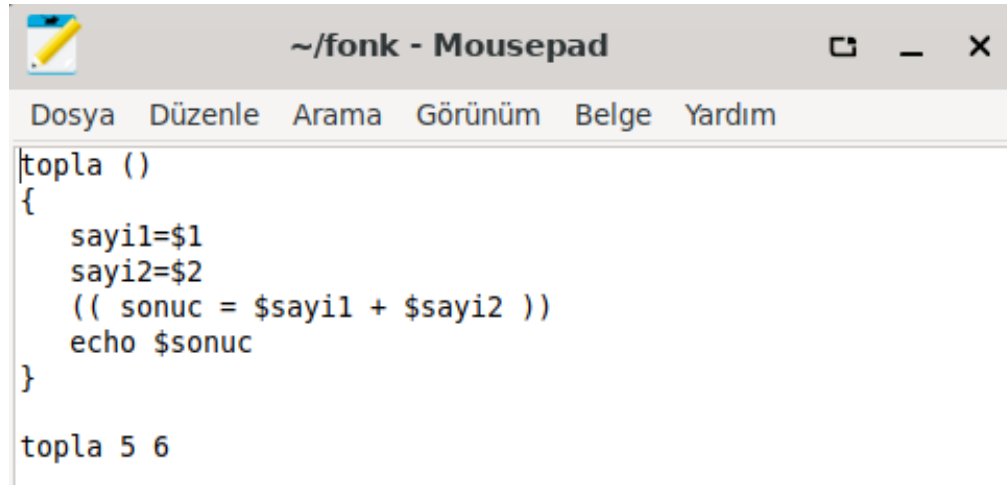
Genel kullanımı:

```
fonksiyon_adi()  
{  
    komutlar  
}  
.....
```

Tanımlanmış bir fonksiyonu çağırırken ise;

```
fonksiyon_adi
```

kullanılır.



```
~/fonk - Mousepad
Dosya  Düzenle  Arama  Görünüm  Belge  Yardım

topla ()
{
    sayi1=$1
    sayi2=$2
    (( sonuc = $sayi1 + $sayi2 ))
    echo $sonuc
}

topla 5 6
```

```
nur@pardus:~$ chmod u+x fonk
nur@pardus:~$ ./fonk
11
nur@pardus:~$
```

Kaynakça

- ▶ Doç. Dr. Mehmet FIRAT, Ders Notları
- ▶ <https://www.gnu.org/>